取扱説明書 - 日本語

ServerView Suite

ServerView でのユーザ管理

中央認証および役割ベースの権限

2010 年 10 月版

DIN EN ISO 9001:2008 に準拠した認証を取得

高い品質とお客様の使いやすさが常に確保されるように、
このマニュアルは、DIN EN ISO 9001:2008
基準の要件に準拠した品質管理システムの規定を
満たすように作成されました。

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

著作権および商標

目次

1 はじめに

このマニュアルでは、ServerView Suite のユーザ管理およびセキュリティアーキテクチャのベースとなる権限および認証のコンセプトについて説明します。

1.1 権限および認証のコンセプト

ServerView Suite のユーザ管理およびセキュリティアーキテクチャは、以下の 3 つの基本コンセプトに基づいています。

– LDAP ディレクトリサービスを使用するグローバルユーザ管理

– 役割ベースのアクセス制御（RBAC）

– 中央認証サービス（CAS）に基づくシングルサイオン（SSO）

LDAP ディレクトリサービスを使用するグローバルユーザ管理

ユーザは、ディレクトリサービスにより、すべての関連する中央管理用サーバに対して一元的に保存および管理されます。ディレクトリサービスは、権限および認証に必要なすべてのデータを提供します。

Sun の OpenDS や、すでに動作している設定済みのディレクトリサービス（Microsoft Active Directory など）などの、ServerView Operations Manager の固有の事前設定されているディレクトリサービスを使用するオプションがあります。

役割ベースのアクセス制御（RBAC）

役割ベースのアクセス制御（RBAC）では、一連のユーザ役割（セキュリティの役割）を定義することにより、アクセス制御を管理します。1 つまたは複数の役割を各ユーザに割り当て、1 つまたは複数のユーザ権限を各役割に割り当てます。

RBAC では、タスク指向の権限プロファイルを各役割に割り当てることにより、ユーザのセキュリティコンセプトとユーザの組織構造を連携させることができます。

RBAC は、ServerView Operations Manager のインストール時に自動的にインストールされる、OpenDS ディレクトリサービスにすでに実装されています。Active Directory などのすでに設定されているディレクトリサービスを使用する場合、そこに補足的に ServerView 固有の権限をインポートできます。続いて、関連する権限を持たせるユーザに必要な役割を割り当てることができます。

シングルサインオン（SSO）

ServerView Suite には、個々のコンポーネントにログインするためのシングルサインオン（SSO）機能があります。SSO は、中央認証サービス（CAS : **C**entral **A**uthentication **S**ervice）に基づいています。SSO では、一度だけユーザ認証を受ける必要があります。一度認証に成功すると、どのコンポーネントでもログインを再び要求されることなく、すべての ServerView コンポーネントにアクセスできます。

1.2 このマニュアルの対象ユーザ

本書は、システム管理者およびネットワーク管理者、ハードウェアおよびソフトウェアの基礎知識のあるサービス技術担当者を対象としています。このマニュアルでは ServerView Suite の権限および認証コンセプトの概要について紹介し、ServerView ユーザ管理のセットアップ手順、および ServerView ユーザ管理をユーザの IT 環境における既存のユーザ管理に統合する手順について詳しく説明します。

1.3 マニュアルの構造

このマニュアルでは、以下のトピックについて説明します。

- **第 2 章 : ユーザ管理およびセキュリティアーキテクチャ（概要）**

  この章では、ServerView Suite の権限および認証コンセプトの概要について紹介します。

- **第 3 章 : LDAP ディレクトリサービスを使用する ServerView ユーザ管理**

  この章では、以下のトピックについて説明します。

  - ディレクトリサービスアクセスの設定
  - OpenDS を使用する ServerView ユーザ管理
  - ServerView ユーザ管理の Microsoft Active Directory への統合

- **第 4 章 : 認証用の SSL 証明書の管理**

  この章では、以下のトピックについて説明します。

  - SSL 証明書の管理（概要）
  - 集中管理サーバ（CMS）での SSL 証明書の管理
  - RBAC およびクライアント認証用の管理対象ノードの準備

- **第 5 章 : Operations Manager へのアクセスに関する役割ベースの許可**

  この章では、事前定義されている ServerView ユーザ役割によって付与される権限に関する詳細情報を説明します。

1.4 ServerView Suite のマニュアル

ServerView Suite のマニュアルは、各サーバシステムに付属の ServerView Suite DVD 2 に収録されています。

また、マニュアルはインターネットから無償でダウンロードすることもできます。オンラインマニュアルは、*http://manuals.ts.fujitsu.com* の *Industry standard servers* のリンク先からダウンロードできます。

1.5 表記規則

このマニュアルでは以下の表記規則を使用します。

⚠ **注意**	この記号は、人的傷害、データ損失、機材破損の危険性を示しています。
ℹ	この記号は、重要な情報やヒントを強調しています。
►	この記号は、操作を続行するために行わなければならない手順を示しています。
斜体	コマンド、ファイル名、およびパス名は、*斜体* で表記されています。
固定フォント	システム出力は、固定フォントで表記されています。
太字の固定フォント	キーボードから入力する必要のあるコマンドは、太字の固定フォントで表記されています。
<abc>	山カッコは、実数値に置き換えられる変数を囲っています。
[パラメータ]	大括弧は、オプション（任意指定）パラメータとオプションを示すために使用されます。
[Key symbols]	キーは、キーボード上の該当するキーを表しています。また、大文字を入力する必要がある場合は、シフトキーも表示されています。 例：大文字 A の場合、[SHIFT] - [A] 2 つのキーを同時に押す必要がある場合は、それぞれのキー記号の間にハイフンが表示されています。

表 1: 本書の表記

このマニュアル内のテキストまたはテキストの項への参照は、章または項の見出しと、章または項の開始ページで示します。

画面出力

画面出力は、使用するシステムに一部依存するため、細部がユーザのシステムに表示される出力と正確に一致しない場合があります。また、使用可能なメニュー項目がシステムによって異なる場合もあります。

2　ユーザ管理およびセキュリティアーキテクチャ（概要）

ServerView Suite のユーザ管理およびセキュリティアーキテクチャに提供される権限および認証のコンセプトは、以下の 3 つの基本コンセプトに基づいています。

– 13 ページ の「LDAP ディレクトリサービスを使用するグローバルユーザ管理」:

  ユーザ名は、ディレクトリサービスを使用して、すべての関連するプラットフォームに対して一元的に保存および管理されます。ディレクトリサービスは、権限および認証に必要なすべてのデータを提供します。

– 15 ページ の「役割ベースのアクセス制御（RBAC）」:

  役割ベースのアクセス制御（RBAC）では、ユーザ役割（セキュリティ役割）を使用して権限を割り当てることにより、ユーザ権限を管理します。この場合、各役割に固有のタスク指向の権限プロファイルを定義します。

– 17 ページ の「CAS サービスを使用するシングルサイオン（SSO）」:

  SSO を使用する場合、一度ログインすると続いて「SSO ドメイン」に参加するすべてのシステムおよびサーバにアクセスでき、その各々に再びログインを要求されることはありません。「SSO ドメイン」は、同じ CAS サービスを使用して、認証を行うすべてのシステムで構成されます。

以降の項では、これらのコンセプトについてより詳しく説明します。

2.1 前提条件

ServerView Suite ユーザ管理およびセキュリティアーキテクチャには以下のソフトウェアが必要です。

– JBoss Web サーバ

  バージョン 5.0 以降、ServerView Operations Manager では JBoss Web サーバを使用します。必要なファイルは ServerView Operations Manager software と共に自動的にインストールされます。

  JBoss は、*ServerView JBoss Applications Server 5.1 と呼ばれる独立したサービスとして設定します。*サービスは次の方法で開始 / 停止できます。

  – Windows システムの場合 : Windows の「スタート」メニューからなど

    「スタート」-「コントロールパネル」-「管理ツール」-「サービス」

  – Linux システムの場合 : `service sv_jboss start` および `service sv_jboss stop` コマンドを使用

– LDAP ディレクトリサービス

  ServerView Operations Manager のインストール時に、ServerView Operations Manager の内部で使用される OpenDS ディレクトリサービスを使用するか、既存のディレクトリサービス（Microsoft Active Directory など）を使用するかを選択できます。

– 中央認証サービス（CAS）

  シングルサインオン（SSO）機能には CAS サービスが必要です。CAS サービスはユーザ認証情報をサーバ側にキャッシュし、ユーザが異なるサービスを要求すると、ユーザに認識されない方法でユーザ認証を行います。

  CAS は ServerView Operations Manager ソフトウェアと共に自動的にインストールされます。

上記コンポーネントを含む ServerView Operations Manager のインストール方法については、『ServerView Operations Manager - Installation under Windows』および『ServerView Operations Manager - Installation under Linux』を参照してください。

2.2 LDAP ディレクトリサービスを使用するグローバルユーザ管理

ServerView Suite のグローバルユーザ管理では、すべての中央管理用サーバ（CMS）のユーザを LDAP ディレクトリサービスのディレクトリに一元的に保存します。これにより、ユーザを中央サーバで管理することができます。そのため、これらのユーザは、ネットワークのこのサーバに接続されるすべての CMS および iRMC S2 で使用できます。

また、CMS のディレクトリサービスを使用すると、同じユーザ ID で CMS および管理対象サーバにログオンできます。

ServerView Suite は現在、以下のディレクトリサービスをサポートしています。

– OpenDS（ServerView Suite の場合）

– Microsoft Active Directory（ServerView Suite および iRMC S2 の場合）

i 外部ディレクトリサービスとしては現在、Active Directory のみサポートされています。ServerView Operations Manager のインストール時には、ServerView の内部ディレクトリサービス（OpenDS）を選択するオプションがあります。iRMC S2 は現在 OpenDS をサポートしていませんのでご注意ください。
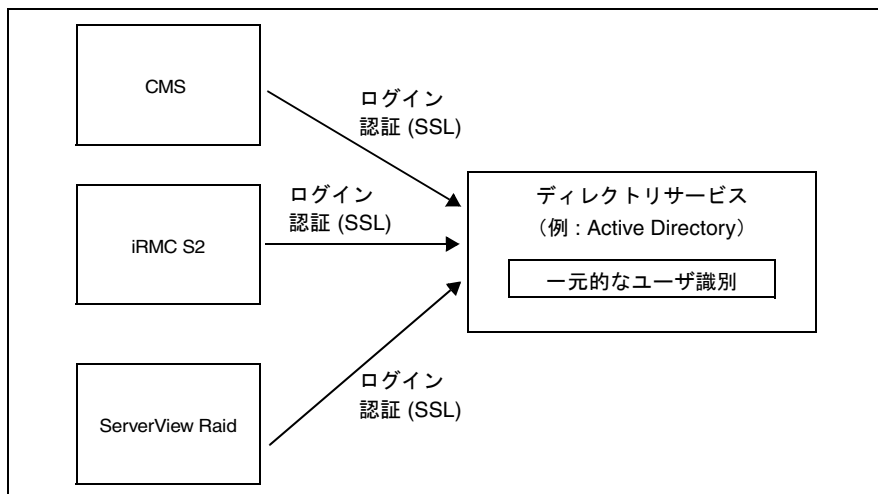
図 1: ServerView Suite のさまざまなコンポーネントでのグローバルユーザの共用

個々の CMS/iRMC S2 などと中央ディレクトリサービスとの間の通信は、TCP/IP プロトコル LDAP（**L**ightweight **D**irectory **A**ccess **P**rotocol）を使用して行われます。LDAP では、最も頻繁に使用されるディレクトリおよびユーザ管理に最も適したディレクトリにアクセスできます。

セキュリティ上の理由のため、LDAP での通信は SSL で保護してください。SSL で保護しない場合、パスワードはプレーンテキストで送信されます。

OpenDS

Operations Manager のインストール時に別のディレクトリサービスを指定しなければ、セットアップ時に Sun の OpenDS が固有のディレクトリサービスとしてインストールされます。サービスは JBoss で Embedded モードで実行されます。そのため、OpenDS は *ServerView JBoss Application Server 5.1* サービスが実行中の場合のみ使用できます。

OpenDS は事前定義されている一連のユーザを提供します。特定のユーザ役割（セキュリティ役割）が各ユーザに割り当てられます。

iRMC S2 は現在 OpenDS をサポートしていません。

既存の設定済みディレクトリサービスの使用

ユーザの IT 環境でディレクトリサービス（Microsoft Active Directory）がすでにユーザ管理に割り当てられている場合、それを ServerView 固有の OpenDS の代わりに使用できます。

2.3 役割ベースのアクセス制御（RBAC）

ServerView Suite のユーザ管理は役割ベースのアクセス制御（RBAC）に基づいているため、ユーザのセキュリティコンセプトとユーザの組織構造を連携させることができます。

ユーザ、ユーザ役割、権限

RBAC では、ユーザに対応する権限を直接割り当てる代わりに、ユーザ役割を使用してユーザへの権限の割り当てを制御します。

– 一連の権限が各ユーザ役割に割り当てられます。各権限セットでは、ServerView Suite のアクティビティにタスク指向の権限プロファイルを定義します。

– 1 つまたは複数の役割が各ユーザに割り当てられます。

ユーザ役割のコンセプトには、次の重要な利点があります。

– 各ユーザまたは各ユーザグループに、個別の権限を個々に割り当てる必要がありません。代わりに、ユーザ役割に権限を割り当てます。

– 権限の構造が変更された場合に、権限をユーザ役割に適応させることのみ必要です。

使用するディレクトリサービスによって、複数の役割を各ユーザに割り当てることができます。この場合、このユーザへの権限は、割り当てられたすべての役割のすべての権限を組み合わせて定義されます。

OpenDS での RBAC の実装

RBAC は、ServerView Operations Manager のインストール時に自動的にインストールされる、OpenDS ディレクトリサービスににすでに実装されています。OpenDS にはユーザ役割の *Administrator*、*Operator* および *Monitor* が事前定義されており、事前定義されているユーザの *administrator*、*operator*、*monitor* にそれぞれ割り当てられます。

i 厳密には、OpenDS は、「cn=Directory Manager」（OpenDS の Directory Manager アカウント）および *svuser*（CAS および ServerView のセキュリティモジュールがディレクトリサービスにアクセスする場合に使用）という、完全に権限が付与されている、特別な目的専用の 2 つの追加のユーザを事前定義します。

個々のユーザ役割によって付与される権限の範囲は、低い方から *Monitor*（最低許可レベル）、*Operator*、*Administrator*（最高許可レベル）です。詳細は、67 ページ の「Operations Manager へのアクセスに関する役割ベースの許可」の章を参照してください。

図 2 に、ユーザ名 *administrator*、*operator*、*monitor* と、対応する役割 *Administrator*、*Operator*、*Monitor* を使用する、役割ベースの割り当てコンセプトを示します。
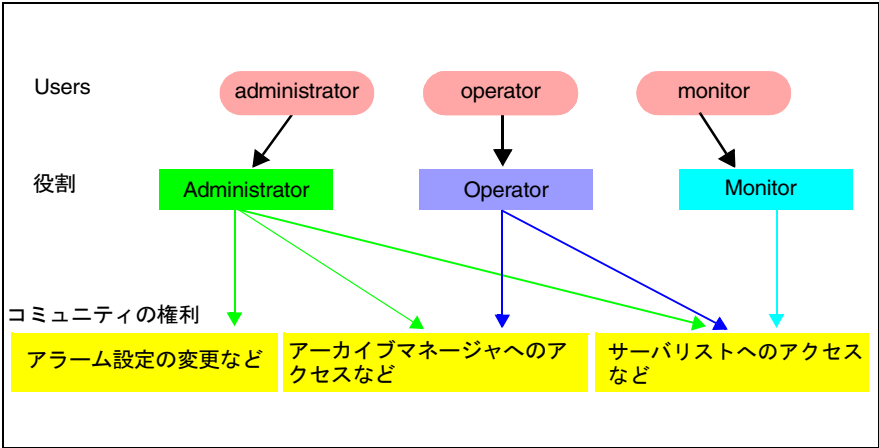
図 2: 役割ベースのユーザ権限の割り当て例

既存の設定済みディレクトリサービスと RBAC との連携

ServerView Suite の RBAC ユーザ管理を、設定済みのディレクトリサービス（Microsoft Active Directory など）に基づく既存の RBAC ユーザ管理に統合することもできます。詳細は、33 ページ の「ServerView ユーザ管理の Microsoft Active Directory への統合」の項を参照してください。

2.4 CAS サービスを使用するシングルサイオン（SSO）

ユーザが個々のコンポーネント（Web サービスなど）にログインできるように、ServerView Suite にはシングルサイオン（SSO）機能があります。ServerView では、中央認証サービス（CAS）を使用して SSO メカニズムを実装し、ユーザからは完全に見えない形で、シングルサインオン手順を処理します。

i SSO を使用するための要件

SSO ドメインに参加するすべてのシステムは、同じ IP アドレス表記を使用して CMS を参照することが必須です。（SSO ドメインは、同じ CAS サービスを使用して認証を行うすべてのシステムで構成されます。）そのため、たとえば「my-cms.my-domain」という名前を使用して ServerView Operations Manager をインストールした場合、これとまったく同じ名前を使用して iRMC S2 の CAS サービスを指定します。そうせずに、「my-cms」のみや my-cms の別の IP アドレスを指定しても、SSO は 2 つのシステム間で有効になりません。。

2.4.1 CAS ベースの SSO アーキテクチャ

SSO アーキテクチャは以下のコンポーネントとアイテムに基づいています。

– 中央認証サービスを提供する CAS サービス
– 「CAS される」任意の ServerView Suite コンポーネントの一部としての CAS クライアント
– サービスチケット（ST : Service Ticket）
– チケット認可チケット（TGT : Ticket Granting Ticket）

中央認証サービス（CAS サービス）によるユーザ認証の管理

CAS サービスは中央ユーザ認証を管理します。この場合、CAS サービスは、管理コンソール（クライアントシステム）のブラウザと、ユーザを管理するディレクトリサービスを仲介します。

CAS クライアントによるサービス要求の遮断およびリダイレクト

CAS クライアントは、「CAS される」任意の ServerView Suite コンポーネントの一部で、ユーザ認証を有効にするためにコンポーネントへの任意の要求を遮断するフィルタです。CAS クライアントはこの要求を CAS サービスにリダイレクトし、続いて CAS サービスがユーザ認証を処理します。

サービスチケット（ST）およびチケット認可チケット（TGT）

ユーザの認証が成功すると、CAS サービスはいわゆるチケット認可チケット（TGT）をユーザに割り当てます、これは技術的に、対応するセキュアなブラウザ Cookie を設定することにより実現します。ServerView Suite コンポーネントの CAS クライアントが HTTPS 要求を CAS サービスにリダイレクトすると、TGT Cookie によりサービスが要求固有のサービスチケット（ST）を作成し、それを追加の要求パラメータを使用して CAS クライアントに返します。CAS クライアントは最初に CAS サービスを直接呼び出して ST を有効にし、次にオリジナルの要求を ServerView Suite コンポーネントに渡します。

チケット認可 Cookie（TGC : Ticket Granting Cookie）

Web ブラウザは、CAS サービスを使用して SSO セッションを確立すると、セキュアな Cookie を CAS サービスに提供します。この Cookie にはチケット認可チケット（TGT）が含まれているため、チケット認可 Cookie（TGT Cookie または TGC）と呼ばれます。

i TGC は、ユーザが CAS をログアウトするかブラウザを閉じると破棄されます。チケット認可チケット Cookie には有効期間があり、CAS サービスのコンフィグレーションファイルに設定されます（事前設定値 : 24 時間）。有効期間は最大 24 時間です。つまり、ユーザは最長 24 時間後にログアウトされます。この最大時間は、インストールされているシステムで変更することはできません。

CAS ベースの SSO が初期のシングルサインオン（SSO）要求を処理する方法

図 3 に、CAS ベースのシングルサインオン（SSO）が初期のシングルサインオン認証を処理する方法を示します。
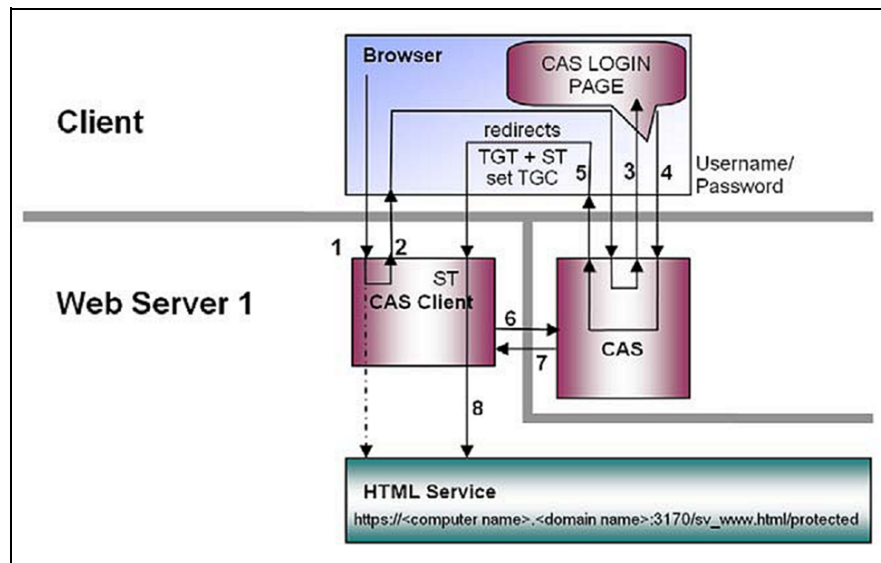
図 3: CAS サービスを使用する SSO アーキテクチャ

説明

1. ユーザは管理コンソールにサービスの URL を入力し、Operations Manager などの ServerView Suite コンポーネントを呼び出します。
2. このユーザ要求は、CAS サービスにリダイレクトされます。
3. CAS サービスは CAS ログインウィンドウを生成し、管理コンソールに表示されます。CAS ログインウィンドウは、ユーザにログイン認証情報（ユーザ名およびパスワード）を要求します。
4. ユーザはログイン認証情報を入力します。
5. CAS サービスはユーザ名およびパスワードを認証し、要求を要求元のコンポーネントにリダイレクトします。また、CAS サービスは TGT Cookie を設定し、ユーザにサービスチケット（ST）およびチケット認可チケット（TGT）を割り当てます。
6. CAS クライアントはサービスチケットを CAS サービスに送信し、検証を要求します。
7. 検証に成功した場合、CAS サービスは、「Service Ticket is ok」という情報とユーザ名を返します。
8. CAS サービスは元の要求に応答します（ステップ 1 を参照）。

CAS ベースの SSO が後続の SSO 要求を処理する方法

サービス（Operations Manager など）へのアクセス認証に成功すると、ログイン認証情報を要求されることなく、ユーザは別のサービス（iRMC S2 Web インターフェースなど）を呼び出すことができます。この場合、CAS サービスは、このユーザが前のログイン手順で設定したチケット認可 Cookie（TGC）を使用して認証を行います。

この TGC が有効なチケット認可 Cookie（TGC）と一致する場合、Web ブラウザが「SSO ドメイン」のサービスに要求を送信するたびに、CAS サービスが自動的にサービスチケット（ST）を発行します。そのため、ユーザは認証情報を要求されずに ServerView Suite コンポーネントにアクセスできます。

2.4.2 ユーザから見たシングルサインオン

SSO では、ユーザが CAS サーバに認証を証明する必要があるのは一度だけです。初めて ServerView Suite コンポーネント（Operations Manager など）にログインすると、CAS サービスによって、ユーザの認証情報（ユーザ名およびパスワード）を要求する別のウィンドウが表示されます。一度認証に成功すると、どのコンポーネントや iRMC S2 でもログインを再び要求されることなく、SSO ドメインのすべての ServerView Suite コンポーネントおよび iRMC S2 にアクセスできます。
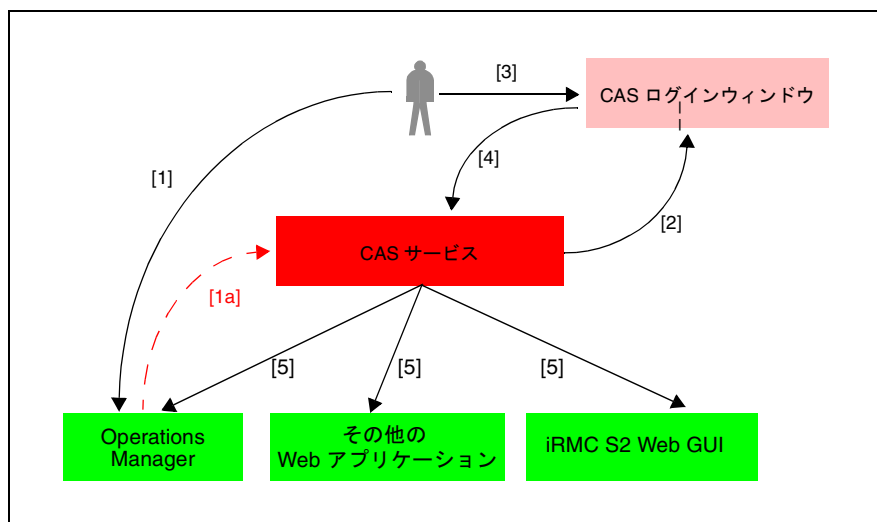
Figure 4: ユーザから見たシングルサインオンの手順

説明

1. ユーザが HTTP 要求を ServerView Suite コンポーネントに送信します

1 a CAS はこの要求を CAS サービスに内部的にリダイレクトします（ユーザには見えません）。

2. CAS サービスが、ユーザにログイン認証情報を要求するログインウィンドウを表示します。

3. ユーザがユーザ名とパスワードを入力し、ユーザの設定を確定します。

4. CAS サービスがこのユーザを認証します。

5. 一度認証に成功すると、再びログインが要求されることなく、使用できます。

3　LDAP ディレクトリサービスを使用する ServerView ユーザ管理

この章では、以下のトピックについて説明します。

– 24 ページ の「ディレクトリサービスアクセスの設定」

– 25 ページ の「OpenDS を使用する ServerView ユーザ管理」

– 33 ページ の「ServerView ユーザ管理の Microsoft Active Directory への統合」

3.1 ディレクトリサービスアクセスの設定

ServerView ユーザ管理の中央認証と役割ベース認証は、どちらも LDAP ディレクトリサービスを使用して一元管理されるデータに基づいて行われます。ディレクトリサービスアクセスは、2 つの別のファイルを使用して制御されます。

– CAS サービスがディレクトリサービスに接続するために必要なデータは、次のファイルに保存されます。

  Windows システムの場合 :

  <ServerView ディレクトリ > ¥jboss ¥server ¥serverview ¥conf ¥casDeployerConfigContext.xml

  Linux システムの場合 :

  /opt/fujitsu/ServerViewSuite/jboss/server/serverview/conf/casDeployerConfigContext.xml

– ServerView セキュリティモジュールがディレクトリサービスに接続するために必要なデータは、次のファイルに保存されます。

  Windows システムの場合 :

  <ServerView ディレクトリ > ¥jboss ¥server ¥serverview ¥conf ¥sv-sec-config.xml

  Linux システムの場合 :

  /opt/fujitsu/ServerViewSuite/jboss/server/serverview/conf/sv-sec-config.xml

i 上記ファイルの変更を有効にするには、JBoss サービスを再起動する必要があります。

3.2 OpenDS を使用する ServerView ユーザ管理

Operations Manager のインストール時に別のディレクトリサービスを指定しなければ、セットアップ時に Sun の OpenDS が固有のディレクトリサービスとしてインストールされます。サービスは JBoss で Embedded モードで実行されます。そのため、OpenDS は *ServerView JBoss Application Server 5.1* サービスが実行中の場合のみ使用できます。

3.2.1 事前定義されているユーザおよび役割

役割ベースのアクセス制御（RBAC）は、OpenDS ディレクトリサービスにすでに実装されています。OpenDS にはユーザ役割の *Administrator*、*Operator* および *Monitor* が事前定義されており、事前定義されているユーザの *administrator*、*operator*、*monitor* にそれぞれ専用に割り当てられます。また、OpenDS には、特別な目的専用の、完全に権限が付与されている 2 つのユーザが事前定義されます。

26 ページ の表 2 に、OpenDS に事前定義されるユーザ名、パスワード、および役割を示します。

注意！

セキュリティを向上させるために、できるだけ早く事前定義されたパスワードを変更してください。パスワードの変更方法の詳細は、27 ページ の「事前定義されたパスワードの変更」の項を参照してください。

個々のユーザ役割に付与される権限の範囲の詳細は、67 ページ の「Operations Manager へのアクセスに関する役割ベースの許可」の章を参照してください。

ユーザ名	パスワード	ユーザ役割	LDAP 識別名 / 説明
./.	admin		cn=Directory Manager,cn=Root DNs,cn=config これは、OpenDS の Directory Manager アカウントです。ルート DN（ルートユーザ）には通常、サーバのすべてのデータへのフルアクセスが付与されます。OpenDS では、ルートユーザはアクセス制御評価をデフォルトでバイパスできるようになります。ルートユーザはサーバ設定にフルアクセスでき、他のほとんどのタイプの操作を行います。 OpenDS では、サーバを複数のルートユーザで設定できます。ルートユーザに付与されるすべての権利は、権限を通じて割り当てられます。
svuser	admin		cn=svuser、ou=users、dc=fujitsu、dc=com このアカウントを使用して、CAS および ServerView のセキュリティモジュールを使用してディレクトリサービスにアクセスします。そのため、対応するコンフィグレーションファイルに関連するデータがあります。それぞれ、*<ServerView ディレクトリ > ¥jboss ¥ server ¥serverview ¥conf ¥ casDeployerContextConfig.xm* および *<ServerView ディレクトリ > ¥jboss ¥server ¥ serverview ¥conf ¥sv-sec-config.xml* など。
administrator	admin	Administrator	cn=ServerView Administrator,ou=users, dc=fujitsu,dc=com Administrator 役割のデフォルトユーザ。
operator	admin	Operator	cn=ServerView Operator,ou=users, dc=fujitsu,dc=com Operator 役割のデフォルトユーザ。
monitor	admin	モニタ	cn=ServerView Monitor,ou=users, dc=fujitsu,dc=com Monitor 役割のデフォルトユーザ。

表 2: OpenDS に事前定義されるユーザ名、パスワード、および役割

3.2.2 事前定義されたパスワードの変更

i 以下の説明では、文字列 "new_mon _pw"、"new_op_pw"、"new_adm_pw"、"new_svu_pw"、"new_dm_pw" で、新しパスワードのプレースホルダを示します。各プレースホルダを、使用する適切なパスワードに置き換えてください。

3.2.2.1 Windows システムでの事前定義されたパスワードの変更

Windows システムの場合、次の手順に従って事前定義されたパスワードを変更します。

1. Windows コマンドプロンプトを開きます。

2. 環境変数 `JAVA_HOME` が the Java Runtime Environment（JRE）のインストールディレクトリに設定されているか確認します。JRE が *C: ¥Program Files (x86) ¥Java ¥jre6* などの下にインストールされる場合、次のコマンドを入力して変数を設定します。

```
SET JAVA_HOME=C:¥Program Files (x86)¥Java¥jre6
```

3. ディレクトリを *<ServerView ディレクトリ > ¥opends ¥bat* に変更します。

4. 次のコマンドを 1 行で入力して、ServerView Monitor のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=ServerView Monitor,ou=users,dc=fujitsu,dc=com"
-n "new_mon_pw"
```

5. 次のコマンドを 1 行で入力して、ServerView Operator のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=ServerView Operator,ou=users,dc=fujitsu,dc=com"
-n "new_op_pw"
```

6. 次のコマンドを 1 行で入力して、ServerView Administrator のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=ServerView Administrator,ou=users,dc=fujitsu,
dc=com" -n "new_adm_pw"
```

7. 次のコマンドを 1 行で入力して、*svuser* のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=svuser,ou=users,dc=fujitsu,dc=com" -n "new_svu_pw"
```

8. 次のコマンドを 1 行で入力して、OpenDS Administrative User のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=Directory Manager,cn=Root DNs,cn=config"
-n "new_dm_pw" -c "admin"
```

9. 以下の両方のファイルで *svuser* のパスワードを同じ値（ここでは例として "new_svu_pw" とします）に変更します。

   <ServerView ディレクトリ > ¥jboss ¥server ¥serverview ¥conf ¥casDeployerConfigContext.xml

   および

   <ServerView ディレクトリ > ¥jboss ¥server ¥serverview ¥conf ¥sv-sec-config.xml

   次のファイルで必要な変更
   <ServerView ディレクトリ > ¥jboss ¥server ¥serverview ¥conf ¥casDeployerConfigContext.xml 29 ページ の図 5:

```
    ...
    <bean id="contextSource"
class="org.springframework.ldap.core.support.LdapContextSource">
        <property name="anonymousReadOnly" value="false" />
        <property name="userDn" value="cn=svuser,ou=users,dc=fujitsu,dc=com" />
        <property name="password" value="new_svu_pw" />
        <property name="pooled" value="true" />
        <property name="urls">
            <list>
                <value>ldaps://pontresina.servware.abg.fsc.net:636/</value>
            </list>
        </property>
        <property name="baseEnvironmentProperties">
            <map>
                        <!-- leave next line as one line. Otherwise setup
postinstall will fail! -->
                <entry> <key> <value>java.naming.security.protocol</value> </key>
<value>ssl</value> </entry>
                <entry>
                    <key>
                        <value>java.naming.security.authentication</value>
                    </key>
                    <value>simple</value>
                </entry>
            </map>
        </property>
    </bean>
    ...
```

図 5: casDeployerContextConfig.xml ファイルでの svuser のパスワードの変更

次のファイルで必要な変更

<ServerView ディレクトリ> ¥jboss ¥server ¥serverview ¥conf ¥sv-sec-config.xml :

```
   ...
   <ns0:authorization>
       <ns0:ldapServer>
           <ns0:serverUrl>ldaps://pontresina.servware.abg.fsc.net:636</ns0:serverUrl>
           <ns0:baseDN>dc=fujitsu,dc=com</ns0:baseDN>
           <ns0:userSearchBase>ou=users,dc=fujitsu,dc=com</ns0:userSearchBase>
           <ns0:userSearchFilter>uid=%u</ns0:userSearchFilter>
           <ns0:serverViewRDN>OU=SVS</ns0:serverViewRDN>

<ns0:securityPrincipal>cn=svuser,ou=users,dc=fujitsu,dc=com</ns0:securityPrincipal>
           <ns0:securityCredentials>new_svu_pw</ns0:securityCredentials>
       </ns0:ldapServer>
       <ns0:department>CMS</ns0:department>
   </ns0:authorization>
   ...
```

図 6: sv-sec-config.xml ファイルでの svuser のパスワードの変更

10. *ServerView JBoss Application Server 5.1* サービスを再起動して、パスワード設定を有効にします。

3.2.2.2 Linux システムでの事前定義されたパスワードの変更

Linux システムの場合、次の手順に従って事前定義されたパスワードを変更します。

1. コマンドシェルを開きます。
2. 環境変数 `JAVA_HOME` が the Java Runtime Environment（JRE）のインストールディレクトリに設定されているか確認します。JRE が */usr/java/default* などの下にインストールされる場合、次のコマンドを入力して変数を設定します。

```
export JAVA_HOME=/usr/java/default
```

3. ディレクトリを */opt/fujitsu/ServerViewSuite/opends/bin* に変更します。
4. 次のコマンドを 1 行で入力して、ServerView Monitor のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=ServerView Monitor,ou=users,dc=fujitsu,dc=com"
-n "new_mon_pw"
```

5. 次のコマンドを 1 行で入力して、ServerView Operator のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=ServerView Operator,ou=users,dc=fujitsu,dc=com"
-n "new_op_pw"
```

6. 次のコマンドを 1 行で入力して、ServerView Administrator のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=ServerView Administrator,ou=users,dc=fujitsu,dc=
com" -n "new_adm_pw"
```

7. 次のコマンドを 1 行で入力して、*svuser* のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=svuser,ou=users,dc=fujitsu,dc=com" -n "new_svu_pw"
```

8. 次のコマンドを 1 行で入力して、OpenDS Administrative User のパスワードを変更します。

```
ldappasswordmodify -h localhost -p 1473
-D "cn=Directory Manager" -w admin
-a "dn:cn=Directory Manager,cn=Root DNs,cn=config"
-n "new_dm_pw" -c "admin"
```

9. 以下の両方のファイルで *svuser* のパスワードを同じ値（ここでは例として "new_svu_pw" とします）に変更します。

   /opt/fujitsu/ServerViewSuite/jboss/server/serverview/conf/casDeployerConfigContext.xml

   および

   /opt/fujitsu/ServerViewSuite/jboss/server/serverview/conf/sv-sec-config.xml

   次のファイルで必要な変更
   /opt/fujitsu/ServerViewSuite/jboss/server/serverview/conf/casDeployerConfigContext.xml：

```
    ...
    <bean id="contextSource"
 class="org.springframework.ldap.core.support.LdapContextSource">
        <property name="anonymousReadOnly" value="false" />
        <property name="userDn" value="cn=svuser,ou=users,dc=fujitsu,dc=com" />
        <property name="password" value="neu_svu_pw" />
        <property name="pooled" value="true" />
        <property name="urls">
            <list>
                <value>ldaps://pontresina.servware.abg.fsc.net:636/</value>
            </list>
        </property>
        <property name="baseEnvironmentProperties">
            <map>
                        <!-- leave next line as one line. Otherwise setup
 postinstall will fail! -->
                <entry> <key> <value>java.naming.security.protocol</value> </key>
 <value>ssl</value> </entry>
                <entry>
                    <key>
                        <value>java.naming.security.authentication</value>
                    </key>
                    <value>simple</value>
                </entry>
            </map>
        </property>
    </bean>
    ...
```

図 7: casDeployerContextConfig.xml ファイルでの svuser のパスワードの変更

次のファイルで必要な変更
/opt/fujitsu/ServerViewSuite/jboss/server/serverview/conf/sv-sec-config.xml：

```
   ...
    <ns0:authorization>
        <ns0:ldapServer>
            <ns0:serverUrl>ldaps://pontresina.servware.abg.fsc.net:636</ns0:serverUrl>
            <ns0:baseDN>dc=fujitsu,dc=com</ns0:baseDN>
            <ns0:userSearchBase>ou=users,dc=fujitsu,dc=com</ns0:userSearchBase>
            <ns0:userSearchFilter>uid=%u</ns0:userSearchFilter>
            <ns0:serverViewRDN>OU=SVS</ns0:serverViewRDN>

<ns0:securityPrincipal>cn=svuser,ou=users,dc=fujitsu,dc=com</ns0:securityPrincipal>
            <ns0:securityCredentials>new_svu_pw</ns0:securityCredentials>
        </ns0:ldapServer>
        <ns0:department>CMS</ns0:department>
    </ns0:authorization>
   ...
```

図 8: sv-config.xml ファイルでの svuser のパスワードの変更

10. ServerView JBoss サービスを再起動して、パスワード設定を有効にします。

3.3 ServerView ユーザ管理の Microsoft Active Directory への統合

Microsoft Active Directory を使用して ServerView ユーザ管理を操作する前に、以下の予備手順に従います。

1. ServerView Suite の役割定義（*Administrator*、*Operator*、*Monitor*。25 ページ を参照）を Active Directory にインポートします。
2. 役割をユーザに割り当てます。

両方の手順については、以下で詳しく説明します。

前提条件：

Active Directory での統合には、ServerView 固有の構造を含む LDIF（Lightweight Directory Interchange Format）ファイルが必要です。

Operations Manager がインストールされている CMS の次のディレクトリに、必要な LDIF ファイルがあります。

– Windows システムの場合 :

<ServerView ディレクトリ> ¥svcommon ¥files ¥SVActiveDirectory.ldif

– Linux システムの場合 :

/opt/fujitsu/ServerViewSuite/svcommon/files/SVActiveDirectory.ldif

以下の手順に従います。

1. **ServerView のユーザ役割定義をインポートします。**

   a) Active Directory を実行している Windows システムの一時ディレクトリに *SVActiveDirectory.ldif* ファイルをコピーします。

   b) Windows コマンドプロンプトを開き、*SVActiveDirectory.ldif* ファイルを含むディレクトリに移動します。

c) Microsoft の *ldifde* ツールを使用して LDIF ファイルをインポートします。

```
ldifde -i -e -k -f SVActiveDirectory.ldif
```

> i *ldifde* ツールがシステムの *PATH 変数*に含まれていない場合は、*%WINDIR% ¥system32* ディレクトリにあります。

> i 必要に応じて、既存の LDAP 構造を新しい権限に追加します。ただし、既存のエントリが影響を受けることはありません。

これで、追加した権限および役割が Active Directory GUI に表示されます。
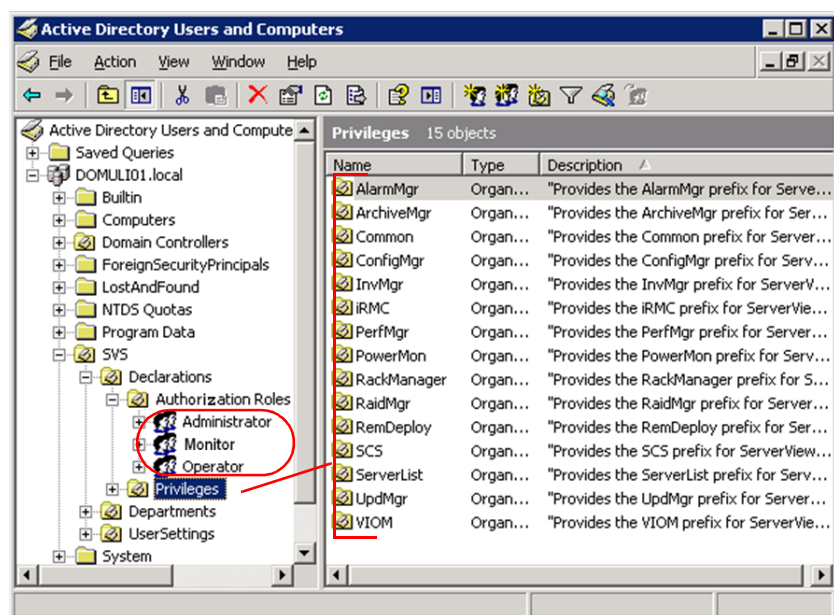
図 9: Active Directory GUI に表示される追加した権限およびユーザ役割

2. ユーザ役割をユーザに割り当てます。

i 以下で説明する手順では、例として、*Monitor* 役割を、Active Directory ドメイン「DOMULI01」でのユーザログイン名が「NYBak」の、「John Baker」に割り当てると想定します。

a) CMS で「スタート」-「コントロールパネル」-「管理ツール」

b) GUI のツリー構造で、*SVS* ノードから *Departments* ノードに移動します。「Departments」の *CMS* および *DEFAULT* を展開します（図 10 を参照）。
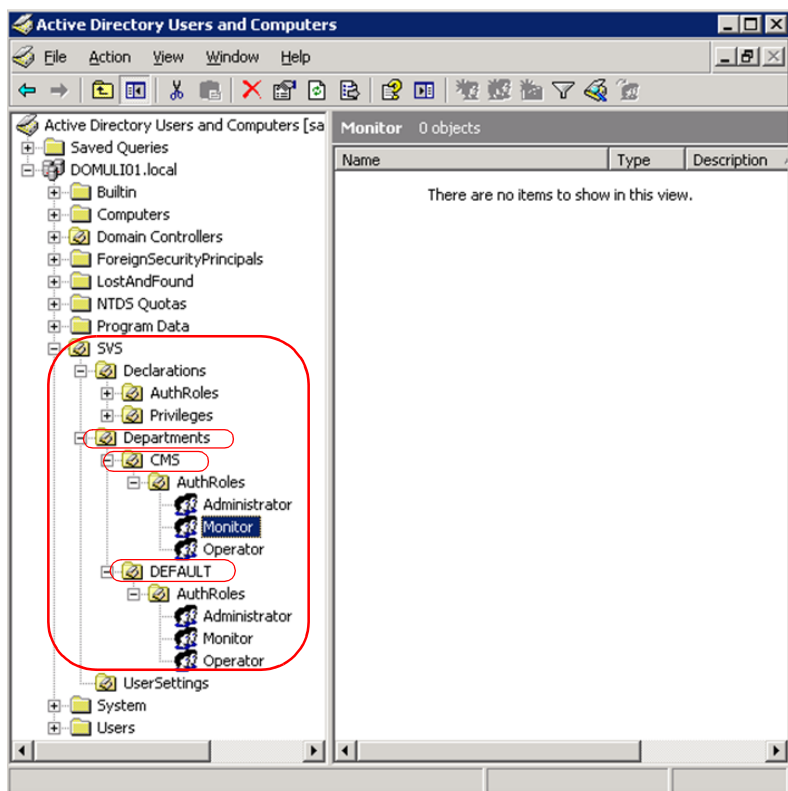
図 10: Monitor 役割をユーザ（John Baker）に割り当てます。

c) 「*SVS*」-「*Departments*」-「*CMS*」-「*AuthorizationRoles* で、*Monitor* を右クリックして*プロパティ*を選択します。

Operator 役割の「プロパティ」ダイアログが表示されます。
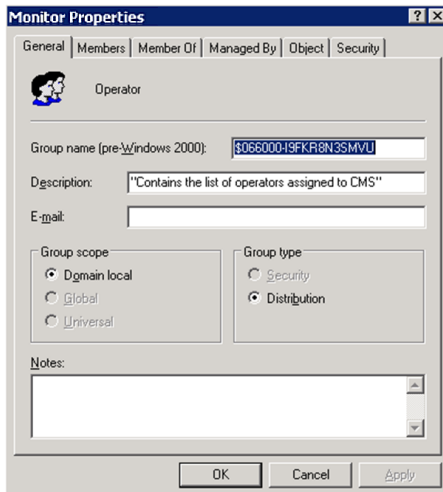
図 11: Monitor 役割の「プロパティ」ダイアログ

d) *メンバー*タブを選択して*追加…*をクリックします。
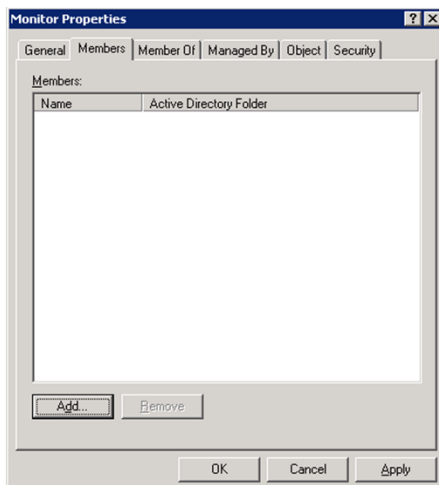
図 12: Monitor の「プロパティ」ダイアログ - 「メンバー」タブ

*ユーザー、連絡先、コンピュータまたはグループの選択*ダイアログが表示されます。
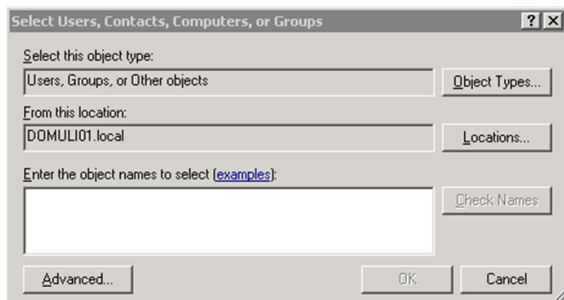
図 13: 「ユーザー、連絡先、コンピュータまたはグループの選択」ダイアログ

「Users」の検索を制限する場合は*オブジェクトの種類*をクリックして *Groups* の選択を解除します。

e) *詳細設定*をクリックします。
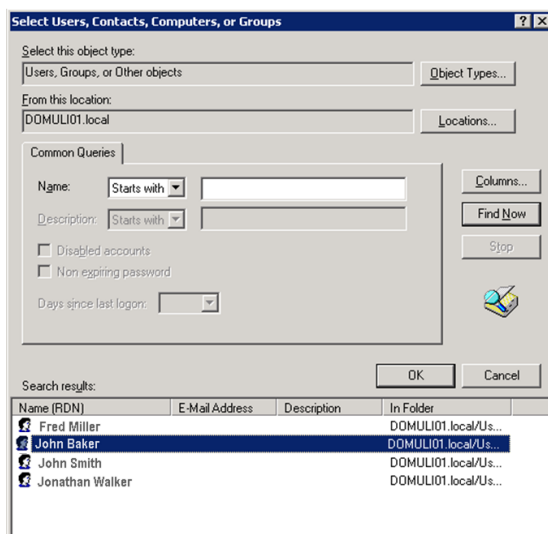
図 14: 必要なユーザを選択します。

i *検索結果*リストで*ログイン名*列を選択してから、*今すぐ検索*をクリックすると、*名前*を制限して検索を迅速に行うことができます。

f) 必要なユーザを選択して *OK* をクリックします。

これで、ユーザ「Baker」が上位ダイアログの*オブジェクト名*リストに表示されます。
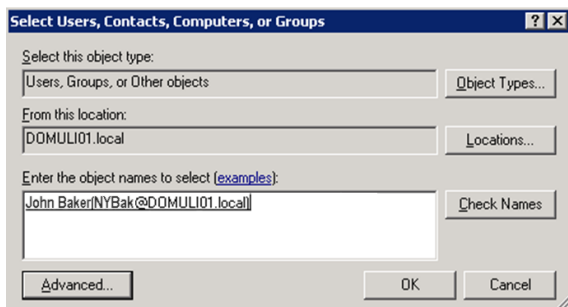
図 15: ユーザー、連絡先、コンピュータまたはグループの選択」ダイアログ : ユーザ「Baker」の表示

g) *OK* をクリックします。

これで、ユーザ「Baker」が *Monitor のプロパティ*ダイアログの*メンバー*タブに表示されます。
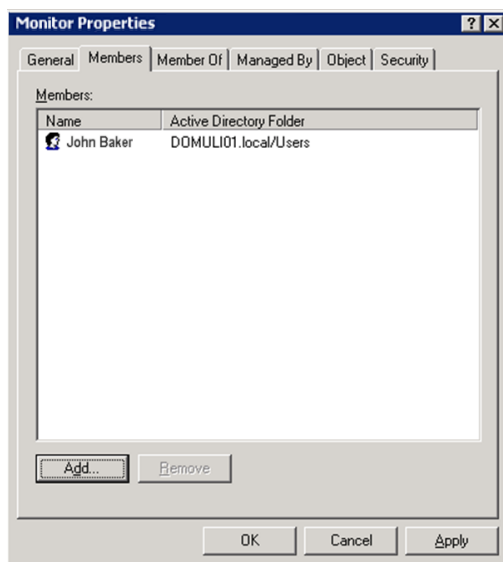
図 16: Monitor の「プロパティ」ダイアログ - 「メンバー」タブ : ユーザ「Baker」の表示

h) *c* ～ *g* の手順を「Department」の *DEFAULT* に繰り返します。

これで、ユーザ「John Baker」は、ユーザ名 NYBak で Operations Manager にログインできます。Baker は、*Monitor* 役割の権限によって許可されたすべての機能を実行できます。

4 認証用の SSL 証明書の管理

Web ブラウザおよび管理対象ノードと通信するために、CMS ではセキュアな SSL 接続で公開鍵インフラストラクチャ（PKI）を使用します。

この章では、以下のトピックについて説明します。

– 42 ページ の「SSL 証明書の管理（概要）」

– 44 ページ の「CMS での SSL 証明書の管理」

– 53 ページ の「RBAC およびクライアント認証用の管理対象ノードの準備」

4.1 SSL 証明書の管理（概要）

Web ブラウザおよび管理対象ノードと通信するために、CMS ではセキュアな SSL 接続で公開鍵インフラストラクチャ（PKI）を使用します。

CMS が自身をサーバ認証により Web サーバで認証する

Web ブラウザは通常、HTTPS 接続（セキュアな SSL 接続）を使用して中央管理用サーバ（CMS）と通信します。そのため、CMS の JBoss Web サーバには、サーバ認証により自身を Web ブラウザに認証するために証明書（X.509 証明書）が必要です。X.509 証明書には、JBoss Web サーバを識別するために必要なすべての情報と、JBoss Web サーバの公開鍵が含まれています。

詳細は、44 ページ の「CMS での SSL 証明書の管理」の項を参照してください。

CMS が自身をクライアント認証により管理対象ノードで認証する

RBAC 機能を使用する管理対象ノード（PRIMERGY サーバなど）には、X.509 証明書ベースのクライアント認証が必要です。そのため CMS は、管理対象ノードに接続するときに、自身を認証する必要があります。クライアント認証により、管理対象ノードが、信頼されない CMS または CMS で実行中の権限のないアプリケーションからアクセスされることを防ぎます。

クライアント認証は、信頼される CMS の証明書があらかじめ管理対象ノードにインストールされていることを前提とします。

詳細は、53 ページ の「RBAC およびクライアント認証用の管理対象ノードの準備」の項を参照してください。

SSL 公開鍵およびセキュリティインタセプタコンフィグレーションファイル

以下のファイルが Operations Manager セットアップ時に自動的に生成されます。

- *< システム名 >.scs.pem*

  PEM 形式の自己署名証明書。PEM ファイルにも公開鍵が含まれます。

  CMS は、以下の目的に *< システム名 >.scs.pem* ファイルを使用します。

  – CMS に接続する Web ブラウザへのサーバ認証。

  – RBAC 機能を使用する管理対象ノードへのクライアント認証。クライアント認証を行うには、*< システム名 >.scs.pem* ファイルを管理対象ノードにインストールする必要があります。

- *< システム名 >.scs.xml*

  セキュリティインタセプタのコンフィグレーションファイル このファイルは、RBAC 検証呼び出しのために内部で使用されます。管理対象ノードで RBAC 機能を有効にするには、*< システム名 >.scs.xml* ファイルを管理対象ノードにインストールする必要があります。

Operations Manager セットアップで CMS の次のディレクトリに両方のファイルがインストールされます。

– *<ServerView ディレクトリ > ¥svcommon ¥data ¥download ¥pki*（Windows システムの場合）

– */opt/fujitsu/ServerViewSuite/svcommon/data/download/pki*（Linux システムの場合）

> **i** 以降、*< システム名 >.scs.pem* および *< システム名 >.scs.xml* は証明書ファイルと略記します。

鍵ペア（keystore ファイルおよび truststore ファイル）の管理

JBoss Web サーバの Java ベースの鍵と証明書の管理では、2 つのファイルを使用して鍵ペアと証明書を管理します。*keystore* ファイルには、JBoss Web サーバによってサーバ固有の鍵ペアが保存されます。*truststore* ファイルには、JBoss Web サーバが信頼できると評価するすべての証明書が含まれます。

keytool ユーティリティ（*keystore* ファイルと *truststore* ファイルを処理するために参照（45 ページを参照））を使用します。

4.2 CMS での SSL 証明書の管理

JBoss Web サーバと通信するために、Web ブラウザは通常、HTTPS 接続（セキュアな SSL 接続）を使用します。そのため JBoss Web サーバには、自身を Web ブラウザで認証するために証明書（X.509 証明書）が必要です。X.509 証明書には、JBoss Web サーバを識別するために必要なすべての情報と、JBoss Web サーバの公開鍵が含まれています。

4.2.1 セットアップ時の自己署名証明書自動作成

PEM 形式の自己署名証明書（*< システム名 >.scs.pem*）は、Operations Manager セットアップ時に（ローカル）JBoss Web サーバに自動的に作成されます。

セットアップで *< システム名 >.scs.pem* が次のディレクトリにインストールされます。

– *<ServerView ディレクトリ > ¥svcommon ¥data ¥download ¥pki*
（Windows システムの場合）

– */opt/fujitsu/ServerViewSuite/svcommon/data/download/pki*
（Linux システムの場合）

自己署名証明書を使用する場合に、ユーザが弊社固有の認証局（CA）の設定や、外部 CA への証明書署名要求（CSR : Certificate Signing Request）の発行に関与することはありません。

i JBoss サーバが自己署名証明書を使用する場合 :
JBoss Web サーバに接続すると、Web ブラウザが証明書エラーを発行し、処理手順を指示します。

可用性が明確なため、自己署名証明書はテスト環境に最適です。ただし、Operations Manager を使用する運用サーバ管理に典型的な高レベルの安全要件を満たすには、信頼される認証局によって署名された証明書（CA 証明書）を使用することを推奨します。

4.2.2 CA 証明書の作成

証明書は、証明書に指定された組織の身元が確認された時点で、中央の認証元である認証局（CA）の秘密鍵を使用して証明書に署名することにより、CA によって発行されます。署名は証明書に含まれ、クライアントが証明書の信頼性を確認できるように、接続セットアップ時に公開されます。

CA 証明書を作成するには次の手順を行う必要があります。

1. *openssl* ツールなどを使用して、証明書署名要求（CSR、ここでは *certrq.pem*）を作成します。

   ```
   openssl req -new -keyout client-key.pem -out certreq.pem
            -days 365
   ```

2. CSR を CA に送信します。

   CA が PEM 形式（*certreply.pem* など）の署名された証明書（証明書応答）を返します。

3. 署名された証明書をファイルに保存します。

4. 署名された証明書を確認します。

4.2.3 証明書と鍵を管理するためのソフトウェアツール

証明書および関連する鍵を管理するには、以下のソフトウェアツールが必要です。

– *openssl*

  openssl ツールは、Shining Light Productions の Web サイト（*http://www.slproweb.com*）などからインターネット経由でダウンロードできます。この代わりに、Cygwin 環境（*http://www.cygwin.com*）のインストールも推奨します。

– *keytool*

  keytool は *http://java.sun.com* からダウンロードできます。*keytool* は Java 仮想マシンとは別にインストールされるため、ユーティリティはデフォルトで中央管理用サーバに保存されます。

  – Windows システム : *C: ¥Program Files (x86) ¥Java ¥jre6 ¥bin など*

  – Linux システム : */usr/java/default/bin*

4.2.4 中央管理用サーバ（CMS）での証明書の交換

この項では、別の証明書に交換する場合に必要な手順について説明します。

前提条件：

下記の手順を行うには、以下のことが必要です。

– 必要なソフトウェア : *openssl*、*keytool*（45 ページを参照）。

  また、次の説明では、*keytool* を含むディレクトリが *PATH* 変数の一部であることを前提とします。

– 署名された CA 証明書（ここでは *certreply.pem*）および秘密鍵（ここでは *privkey.pem*）が使用可能である必要があります。

中央管理用サーバで証明書を交換した後、管理対象ノードでも証明書を交換する必要があります（Windows 管理対象ノードの場合は 58 ページ、Linux/VMware 管理対象ノードの場合は 59 ページを参照）。これにより、CMS は管理対象ノードでの認証を継続できます。

4.2.4.1 Windows システムでの証明書の交換

以下の手順に従います。

1. JBoss サービスを停止します（12 ページを参照）。
2. *keystore* ファイルを削除します。
   a) Windows コマンドプロンプトを開きます。
   b) *<ServerView ディレクトリ > ¥jboss ¥server ¥serverview ¥conf ¥pki* ディレクトリに移動します。
   c) *keystore* ファイルを削除するかファイル名を変更します。
3. 署名された CA 証明書（ここでは *certreply.pem*）を新しい *keystore* ファイルにインポートし、公開鍵（ここでは *keystore.p12*）をエクスポートします。

```
openssl pkcs12 -export -in certreply.pem -inkey privkey.pem
     -passout pass:%STOREPASS% -out keystore.p12 -name
svs_cms
```

> i プレースホルダ %STOREPASS% を、Operations Manager セットアップ時に指定した *keystore* パスワードに置き換えます。セットアップが完了すると、*<ServerView ディレクトリ > ¥jboss ¥bin ¥ run.conf.bat* ファイルの最後にある、Java オプション *javax.net.ssl.keyStorePassword* が定義される行に、*keystore* パスワードが追加されます。

4. *keystore* ファイルを（再）フォーマットします。

```
keytool -importkeystore -srckeystore keystore.p12
     -destkeystore keystore -srcstoretype PKCS12
     -srcstorepass %STOREPASS% -deststorepass %STOREPASS%
     -destkeypass %KEYPASS% -srcalias svs_cms
    -destalias svs_cms -noprompt -v
```

> i プレースホルダ %KEYPASS% を、Operations Manager セットアップ時に指定した鍵パスワードに置き換えます。セットアップが完了すると、*<ServerView ディレクトリ > ¥jboss ¥bin ¥run.conf.bat* ファイルの最後にある、Java オプション *javax.net.ssl.keyPassword* が定義される行に、*keystore* パスワードが追加されます。

5. 新しい証明書を *truststore* ファイルにインポートします。

   これを最も容易に行うには以下の手順に従います。

   a) JBoss サービスを開始します。
   b) スタートアップが完了するまで待ちます。

c) *<ServerView ディレクトリ > ¥jboss ¥server ¥serverview ¥bin* ディレクトリに移動します。

d) Windows コマンドプロンプトを開いて以下のコマンドを入力します。

```
java -jar install-cert-gui-ComJEE_V0.10.jar -s ..¥conf¥
pki¥cacerts %STOREPASS% < システム FQDN>:3170
```

i 設定済みの外部ディレクトリサービスを使用する場合は、次のコマンドも入力することが必要です。

```
java -jar install-cert-gui-ComJEE_V0.10.jar -s ..¥
conf¥pki¥cacerts %STOREPASS% < システム FQDN>:<port>
```

< システム FQDN>
ご利用のシステムの完全修飾識別名です。

<port>
外部ディレクトリサービスに使用される LDAP ポート
（通常 : 636）

i Java プログラムを呼び出すと、次のメッセージが表示されます。

```
testConnection(tm,pontresina.servware.abg.fsc.net,
3170): SSLException: java.lang.RuntimeException:
Unexpected error:
java.security.InvalidAlgorithmParameterException: the
trustAnchors parameter must be non-empty
```

```
writing to truststore ..¥conf¥pki¥cacerts...
```

これはエラーではなく、新しい証明書がまだ *truststore* ファイルにインポートされていないことを示しているだけです。

e) PEM 形式の *keystore.pem* ファイルを作成します。

以下の手順に従います。

► 次のコマンドを適用します。

```
openssl pkcs12 -in keystore.p12 passin pass:%STOREPASS%
-out keystore.pem -passout pass:%STOREPASS%
```

► テキストエディタで *keystore.pem* ファイルを開き、以下以外のすべてのテキスト行を削除します。

– 「-----」の印のついたヘッダーおよびフッター
– 暗号化されたデータブロック行

► *keystore.pem* ファイルを CMS の次のディレクトリにコピーします。

<ServerView ディレクトリ> ¥jboss ¥server ¥serverview ¥conf ¥pki

f) PEM 形式の証明書ファイル（*< システム名 >.scs.pem*）を作成します。

以下の手順に従います。

► 以下のコマンドを適用します。

```
keytool -exportcert -keystore keystore -storepass
%STOREPASS% -alias svs_cms -file <ÉVÉXÉeÉÄñº>.scs.crt
```

```
openssl -in <ÉVÉXÉeÉÄñº>.scs.crt -inform DER -out
<ÉVÉXÉeÉÄñº>.scs.pem -outform PEM
```

► *< システム名 >.scs.pem* ファイルを CMS の次のディレクトリにコピーします。

<ServerView ディレクトリ> ¥svcommon ¥data ¥download ¥pki

6. JBoss サービスを再起動して、変更を有効にします。

4.2.4.2 Linux システムでの証明書の交換

以下の手順に従います。

1. JBoss サービスを停止します（12 ページを参照）。
2. *keystore* ファイルを削除します。
   a) xterm ターミナルや gnome ターミナルなどの、ターミナルを開きます。
   b) */opt/fujitsu/ServerViewSuite/jboss/server/serverview/conf/pki* ディレクトリに移動します。
   c) *keystore* ファイルを削除するかファイル名を変更します。
3. 署名された CA 証明書（ここでは *certreply.pem*）を新しい *keystore* ファイルにインポートし、公開鍵（ここでは *keystore.p12*）をエクスポートします。

```
openssl pkcs12 -export -in certreply.pem -inkey privkey.pem
     -passout pass:%STOREPASS% -out keystore.p12 -name
svs_cms
```

> i プレースホルダ `%STOREPASS%` を、Operations Manager セットアップ時に指定した *keystore* パスワードに置き換えます。セットアップが完了すると、*/opt/fujitsu/ServerViewSuite/jboss/bin/run.conf* ファイルの最後の、Java オプション *javax.net.ssl.keyStorePassword* が定義される行に、*keyststore* パスワードが追加されます。

4. *keystore* ファイルを（再）フォーマットします。

```
keytool -importkeystore -srckeystore keystore.p12
     -destkeystore keystore -srcstoretype PKCS12
     -srcstorepass %STOREPASS% -deststorepass %STOREPASS%
     -destkeypass %KEYPASS% -srcalias svs_cms
    -destalias svs_cms -noprompt -v
```

> i プレースホルダ `%KEYPASS%` を、Operations Manager セットアップ時に指定した鍵パスワードに置き換えます。セットアップが完了すると、*/opt/fujitsu/ServerViewSuite/jboss/bin/run.conf* ファイルの最後の、Java オプション *javax.net.ssl.keyPassword* が定義される行に、*keyststore* パスワードが追加されます。

5. 新しい証明書を *truststore* ファイルにインポートします。

   これを最も容易に行うには以下の手順に従います。

   a) JBoss サービスを開始します。
   b) スタートアップが完了するまで待ちます。

c) *../../bin* ディレクトリに移動します。

d) ターミナルウィンドウを開いて以下のコマンドを入力します。

```
java -jar install-cert-gui-ComJEE_V0.10.jar -s
../conf/pki/cacerts %STOREPASS% < システム FQDN>:3170
```

> **i** 設定済みの外部ディレクトリサービスを使用する場合は、次のコマンドも入力することが必要です。
>
> ```
> java -jar install-cert-gui-ComJEE_V0.10.jar -s
> ../conf/pki/cacerts %STOREPASS% <systems FQDN>:<port>
> ```

< システム FQDN>
: ご利用のシステムの完全修飾識別名です。

<port>
: 外部ディレクトリサービスに使用される LDAP ポート（通常 : 636）

> **i** Java プログラムを呼び出すと、次のメッセージが表示されます。
>
> ```
> testConnection(tm,pontresina.servware.abg.fsc.net,
> 3170): SSLException: java.lang.RuntimeException:
> Unexpected error:
> java.security.InvalidAlgorithmParameterException: the
> trustAnchors parameter must be non-empty
> ```
>
> ```
> writing to truststore ../conf/pki/cacerts...
> ```
>
> これはエラーではなく、新しい証明書がまだ *truststore* ファイルにインポートされていないことを示しているだけです。

e) PEM 形式の *keystore.pem* ファイルを作成します。

以下の手順に従います。

► 次のコマンドを適用します。

```
openssl pkcs12 -in keystore.p12 passin pass:%STOREPASS%
-out keystore.pem -passout pass:%STOREPASS%
```

► テキストエディタで *keystore.pem* ファイルを開き、以下以外のすべてのテキスト行を削除します。

– 「-----」の印のついたヘッダーおよびフッター
– 暗号化されたデータブロック行

► *keystore.pem* ファイルを CMS の次のディレクトリにコピーします。

/opt/fujitsu/ServerViewSuite/jboss/server/serverview/conf/pki

f) PEM 形式の証明書ファイル（< *システム名* >*.scs.pem*）を作成します。

以下の手順に従います。

► 以下のコマンドを適用します。

```
keytool -exportcert -keystore keystore -storepass
%STOREPASS% -alias svs_cms -file <ÉVÉXÉeÉÄñº>.scs.crt
```

```
openssl -in <ÉVÉXÉeÉÄñº>.scs.crt -inform DER -out
<ÉVÉXÉeÉÄñº>.scs.pem -outform PEM
```

► < *システム名* >*.scs.pem* ファイルを CMS の次のディレクトリにコピーします。

/opt/fujitsu/ServerViewSuite/svcommon/data/download/pki

6. JBoss サービスを再起動して、変更を有効にします。

4.3 RBAC およびクライアント認証用の管理対象ノードの準備

RBAC およびクライアント認証用の管理対象ノードの準備には、次の手順が必要です。

1. 証明書ファイル（< *システム名* >*.scs.pem* および < *システム名* >*.scs.xml*）を管理対象ノードに転送します。
2. 転送したファイルを管理対象ノードにインストールします。

4.3.1 < システム名 >.scs.pem および < システム名 >.scs.xml の管理対象ノードへの転送

CMS での Operations Manager セットアップが正常に終了すると、< *システム名* >*.scs.pem* および < *システム名* >*.scs.xml* は CMS の次のディレクトリに保存されます。

– *<ServerView ディレクトリ > ¥svcommon ¥data ¥download ¥pki*（Windows システムの場合）

– */opt/fujitsu/ServerViewSuite/svcommon/data/download/pki*（Linux システムの場合）

管理対象ノードは「手動で」転送できますが、CMS からダウンロードするほうが容易です。

ファイルをダウンロードするための要件

管理対象ノードで Web ブラウザが使用できること。

ユーザに *Administrator* 役割が割り当てられていること。

ファイルをダウンロードするには、次の手順に従います。

1. 管理対象ノードのブラウザで、次の URL を入力します。

   https:< システム名 >:3170/Download/pki

   （< *システム名* > には CMS の DNS 名または IP アドレスを入力します）

   次のウィンドウが開き、ダウンロードの用意ができているファイルが表示されます。
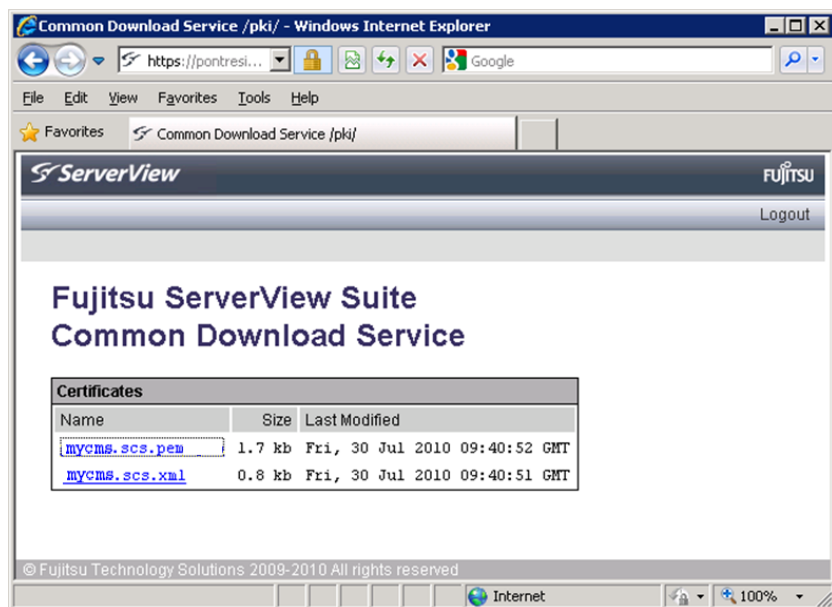
図 17: CMS mycms からの mycms.scs.pem および mycms.scs.xml のダウンロード

2. 各ファイルの対応するリンクを右クリックして、*対象をファイルに保存*を選択してファイルを管理対象ノードに保存します。

4.3.2 Windows システムでの証明書ファイルのインストール

証明書ファイル *< システム名 >.scs.pem* および *< システム名 >.scs.xml* のインストールには、以下のオプションがあります。

– ServerView エージェントと共に証明書ファイルを初期インストールする。

– 証明書ファイルを ServerView エージェントがすでにインストールされている管理対象ノードにインストールする（CMS で対応する交換を行ったために、最初にインストールした自己署名証明書を信頼される CA の証明書に交換しなければならない場合など）。

4.3.2.1 ServerView エージェントと共に証明書ファイルをインストールする

i この場合、ServerView エージェントを実際にインストールする前に、証明書ファイルを管理対象ノードにインストールしします。

次に、Windows システムでの証明書ファイルのインストール方法を説明します。ServerView エージェントのインストール方法の詳細については、『Installation ServerView Agents for Windows』マニュアルの該当する項を参照してください。

圧縮されたセットアップを使用するインストール

以下の手順に従います。

1. 圧縮されたエージェントセットアップファイル（*ServerViewAgents_Win_i386.exe* または *ServerViewAgents_Win_x64.exe*）をネットワーク共有または管理対象ノードのローカルディレクトリにコピーします。

2. セットアップファイルが保存されているディレクトリに、新しいディレクトリ *pki*（「public key infrastructure」の略）を作成します。

3. 証明書ファイル *< システム名 >.scs.pem* および *< システム名 >.scs.xml* を新しい *pki* ディレクトリに転送します。複数の証明書を複数の信頼される CMS に転送することもできます。

4. 圧縮されたセットアップを実行します（詳細については『Installation ServerView Agents for Windows』を参照）。

   ServerView エージェントのセットアップ時に、*pki* ディレクトリのすべての証明書が適切な場所にインストールされます。

解凍されたセットアップを使用するインストール

以下の手順に従います。

1. 解凍されたセットアップファイル *ServerViewAgents_Win_i386.exe* または *ServerViewAgents_Win_x64.exe* をネットワーク共有または管理対象ノードのローカルディレクトリにコピーします。

   Setup.exe、*ServerViewAgents_xxx.msi* およびその他のファイルが作成されます。

2. セットアップファイルが保存されているディレクトリに、新しいディレクトリ *pki*（「public key infrastructure」の略）を作成します。

3. 証明書ファイル *< システム名 >.scs.pem* および *< システム名 >.scs.xml* を新しい *pki* ディレクトリに転送します。複数の証明書を複数の信頼される CMS に転送することもできます。

4. *Setup.exe* を実行します（詳細については『Installation ServerView Agents for Windows』を参照）。

   ServerView エージェントのセットアップ時に、*pki* ディレクトリのすべての証明書が適切な場所にインストールされます。

ServerView Suite DVD 1 からのインストール

i ServerView エージェントおよび証明書は、ServerView Suite DVD 1 から直接インストールできません。

以下の手順に従います。

1. 圧縮または解凍されたエージェントセットアップファイルを、ServerView Suite DVD 1 からネットワーク共有または管理対象ノードのローカルディレクトリにコピーします。

2. セットアップファイルが保存されているディレクトリに、新しいディレクトリ *pki*（「public key infrastructure」の略）を作成します。

3. 証明書ファイル *< システム名 >.scs.pem* および *< システム名 >.scs.xml* を新しい *pki* ディレクトリに転送します。複数の証明書を複数の信頼される CMS に転送することもできます。

4. 圧縮されたセットアップを実行します（詳細については『Installation ServerView Agents for Windows』を参照）。

   ServerView エージェントのセットアップ時に、*pki* ディレクトリのすべての証明書が適切な場所にインストールされます。

4.3.2.2 ServerView エージェントがすでにインストールされている Windows システムでの証明書ファイルのインストール

以下の手順に従います。

1. 管理対象ノードで ServerView Remote Connector Service（SCS）へのパス（以下 *<scsPath>* と略記）を探します。

   デフォルトのパスは次のとおりです。

   – x64 システムの場合 :

     C: ¥Program Files (x86) ¥Fujitsu ¥ServerView Suite ¥Remote Connector

   – i386 システムの場合 :

     C: ¥Program Files ¥Fujitsu ¥ServerView Suite ¥Remote Connector

2. 証明書ファイル *< システム名 >.scs.pem* および *< システム名 >.scs.xml* を SCS 証明書フォルダ *<scsPath> ¥pki* に転送します。

   新しい証明書や変更された証明書は、10 秒以内、または Remote Connector Service の再起動後に SCS によってリロードされます。

4.3.3 Linux または VMware システムでの証明書ファイルのインストール

証明書ファイル < *システム名* >*.scs.pem* および < *システム名* >*.scs.xml* のインストールには、以下のオプションがあります。

– ServerView エージェントと共に証明書ファイルを初期インストールする。

– 証明書ファイルを ServerView エージェントがすでにインストールされている管理対象ノードにインストールする（CMS で対応する交換を行ったために、最初にインストールした自己署名証明書を信頼される CA の証明書に交換しなければならない場合など）。

4.3.3.1 ServerView エージェントと共に証明書ファイルをインストールする

i この場合、実際にシェルコマンドでインストールを開始する前に、証明書ファイルを管理対象ノードにインストールします。

i 次に、Linux または VMware システムでの証明書ファイルのインストール方法を説明します。ServerView エージェントのインストール方法の詳細については、『Installation ServerView Agents for Linux』マニュアルの該当する項を参照してください。

ServerView Suite DVD 1 からのインストール

1. < *システム名* >*.scs.pem* および < *システム名* >*.scs.xml* を */temp* ディレクトリに転送します。

2. 次のコマンドを入力して環境変数 *SV_SCS_INSTALL_TRUSTED* をエクスポートします。

   export SV_SCS_INSTALL_TRUSTED=/tmp

3. 次のコマンドを入力します : *sh srvmagtDVD.sh [-R]*

   証明書ファイル < *システム名* >*.scs.pem* および < *システム名* >*.scs.xml* がインポートされます。

   新しい証明書や変更された証明書は、10 秒以内、または Remote Connector Service の再起動後に SCS によってリロードされます。

ディレクトリからのインストール

1. < *システム名* >*.scs.pem* および < *システム名* >*.scs.xml* を、ServerView エージェントのモジュールを含むローカルディレクトリに転送します。
2. 次のコマンドを入力します。*sh ./srvmagt.sh [option] install*

   証明書ファイル < *システム名* >*.scs.pem* および < *システム名* >*.scs.xml* がインポートされます。

rpm コマンドを使用するインストール

1. < *システム名* >*.scs.pem* および < *システム名* >*.scs.xml* をローカルディレクトリ *<cert dir>* に転送します。
2. 次のコマンドを入力して環境変数 *SV_SCS_INSTALL_TRUSTED* をエクスポートします。

   export SV_SCS_INSTALL_TRUSTED=<cert dir>
3. 次のコマンドを入力します。

   rpm -U ServerViewConnectorService-<scs バージョン >.i386.rpm

   証明書ファイル < *システム名* >*.scs.pem* および < *システム名* >*.scs.xml* がインポートされます。

4.3.3.2 ServerView エージェントがすでにインストールされている Linux/VMware システムでの証明書ファイルのインストール

以下の手順に従います。

1. ターミナルを起動します（*root* として）。
2. 管理対象ノードで ServerView Remote Connector Service（SCS）へのパス（以下 *<scsPath>* と略記）を探します。

   デフォルトのパスは次のとおりです。

   /opt/fujitsu/ServerViewSuite/SCS/pki
3. < *システム名* >*.scs.pem* および < *システム名* >*.scs.xml* をローカルディレクトリに転送します。
4. 次のコマンドを入力します。

   cp -p < *システム名* >*.scs.pem* < *システム名* >*.scs.xml <scsPath>*

   新しい証明書や変更された証明書は、10 秒以内、または Remote Connector Service の再起動後に SCS によってリロードされます。

4.3.4 ServerView Update Manager を使用する証明書のインストール（Windows/ Linux/VMware システム）

前提条件 :

ServerView Update エージェントおよびサーバエージェントはバージョン 5.0 以降である必要があります。

サーバリストに表示される各管理対象ノードに対して、ServerView アップデートマネージャのアップデートメカニズムを使用して、CMS 証明書を管理対象ノードにサーバリストから直接インストールできます。他のアップデートコンポーネントの場合と同様に、アップデートマネージャにより、インストールに使用可能なソフトウェアとして CMS 証明書が提供されます。アップデートジョブを作成して開始することにより、証明書を管理対象ノードに自動的に転送できます。

この場合、CMS について生成された各証明書ファイルは、アップデートマネージャに割り当てられているリポジトリに置く必要があります（パス : *... ¥Tools ¥Certificates*（Windows の場合）および *.../Tools/Certificates*（Linux/VMware の場合）:

– リポジトリの通常の初期設定では、アップデートマネージャの設定ウィザードにより、設定の最後に証明書がリポジトリに自動的に追加されます。

– アップデートインストール時に、該当するインストールスクリプトを実行することにより、証明書がリポジトリに自動的に追加されます。

重要 !

追加されるデータは各 CMS にのみ有効なため、ローカルリポジトリの指定しかできません。

4.3.4.1 管理対象ノードでの ServerView Update Manager を使用した CMS 証明書のインストール（概要）

下記の説明のようにアップデートマネージャのメインウィンドウを使用して、管理対象ノードでの CMS 証明書のインストールを制御できます。

アップデートマネージャの詳細は、『ServerView Update Manager』マニュアルを参照してください。

アップデートマネージャのメインウィンドウの「サーバ詳細」タブ（管理対象ノードに CMS 証明書をインストールする前）

CMS 証明書を管理対象ノードにインストールしない限り、*サーバ詳細*タブでこのノードの*エージェントアクセス*列に、「not certified」と表示されます（図 18 を参照）。

> 管理対象ノードの ServerView Update エージェントおよび ServerView エージェントが 5.0 以降の場合、*サーバ詳細*タブでこのノードの*エージェントアクセス*列に、「restricted」または「unrestricted」と表示されます。
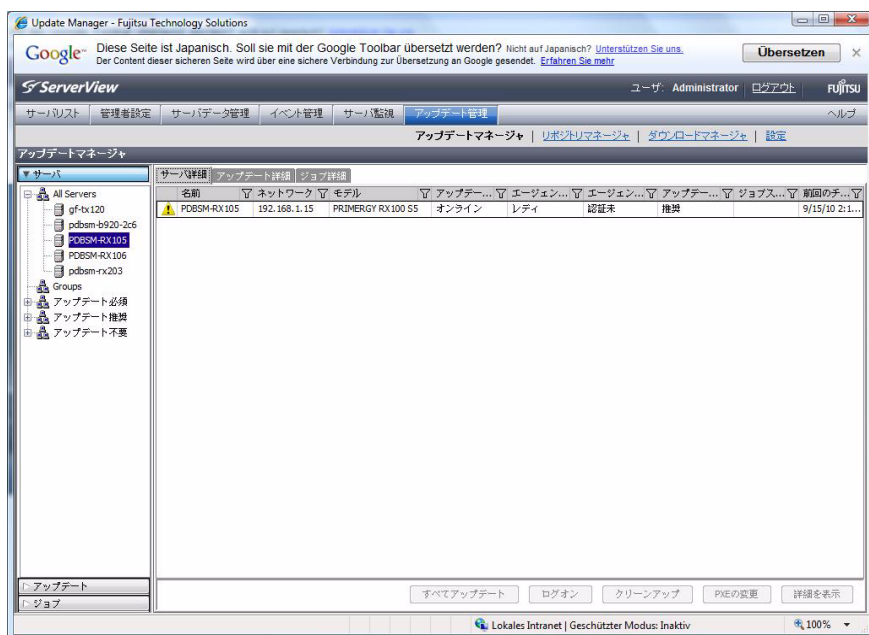
図 18: アップデートマネージャのメインウィンドウ - 「サーバ詳細」タブ（CMS 証明書がまだインストールされていない）

アップデートマネージャのメインウィンドウの「アップデート詳細」タブ（管理対象ノードに CMS 証明書をインストールする前）

*アップデート詳細*タブの*アップグレード*ビューでは、各行に選択したノードの CMS 証明書に関するインストールオプションが示されます（図 19 を参照）。
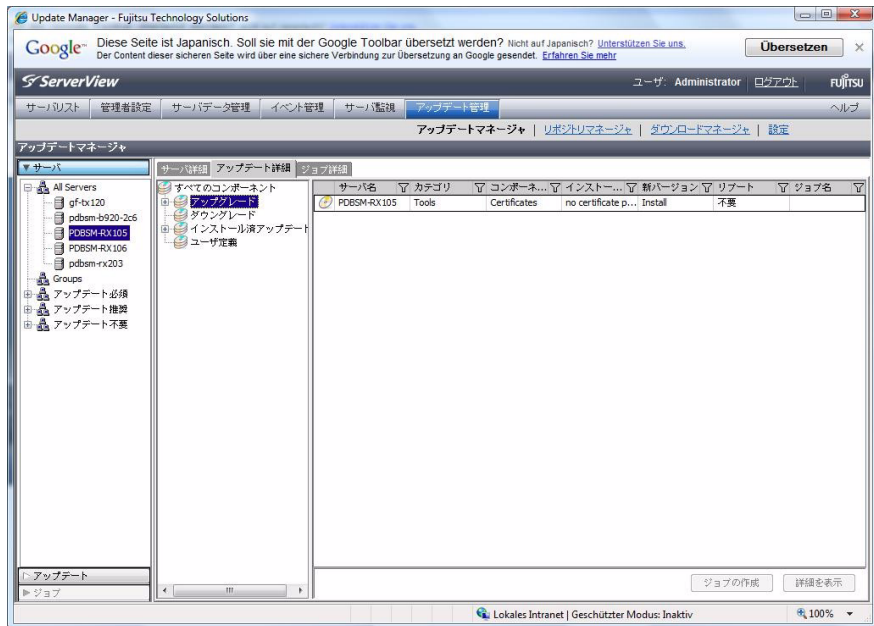
図 19: アップデートマネージャのメインウィンドウ - 「アップデート詳細」タブ
（CMS 証明書がまだインストールされていない）

ここで、管理対象ノードでこのインストールを実行する新しいアップデートジョブを作成して開始できます。（オプションで、アップデートジョブを追加のアップデートコンポーネントで構成することもできます）。アップデートジョブの作成方法の詳細は、『ServerView Update Manager』マニュアルを参照してください。

アップデートマネージャのメインウィンドウの「サーバ詳細」タブ（CMS ウィンドウで CMS 証明書が正常にインストールされた後）

CMS 証明書が正常にインストールされると、*サーバ詳細*タブでこのノードの*エージェントアクセス*列に、「certified」と表示されます（図 20 を参照）。
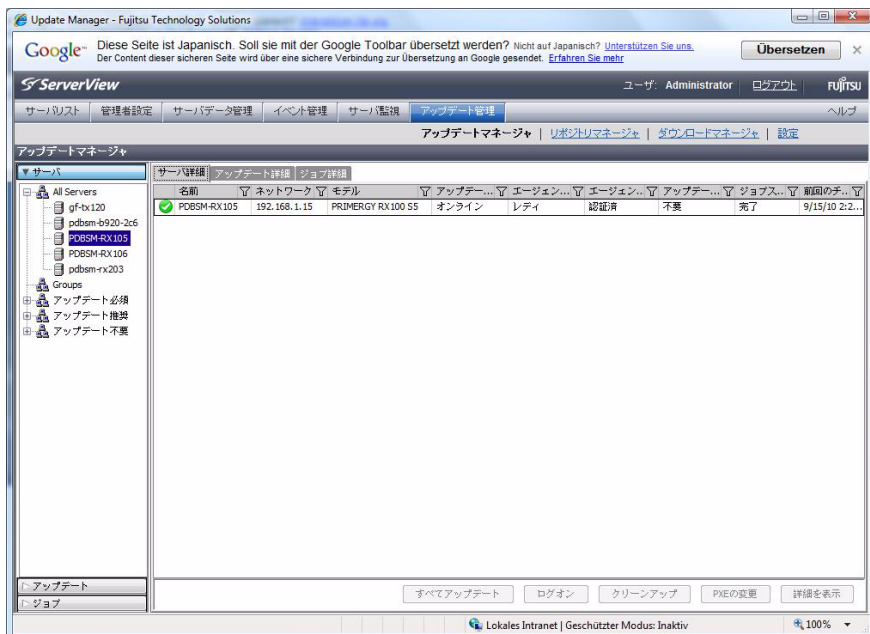
図 20: アップデートマネージャのメインウィンドウ - 「サーバ詳細」タブ
（CMS 証明書が正常にインストールされた）

アップデートマネージャのメインウィンドウの「アップデート詳細」タブ（CMS ウィンドウで CMS 証明書が正常にインストールされた後）

CMS 証明書が管理対象ノードに正常にインストールされると、*アップデート詳細*タブの*インストール済アップデート*ビューに、CMS 証明書が管理対象ノードに正常にインストールされたことが示されます（図 21 を参照）。
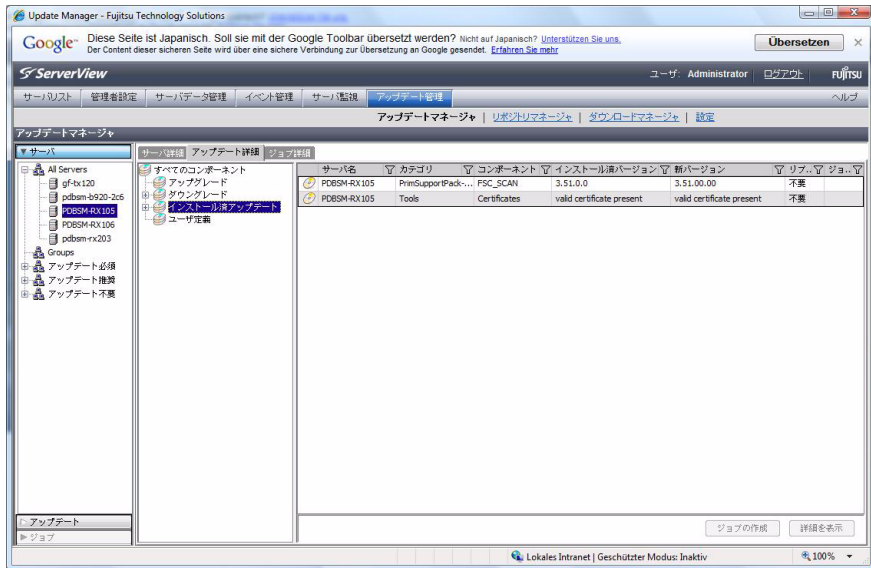
図 21: アップデートマネージャのメインウィンドウ - 「アップデート詳細」タブ（CMS 証明書が正常にインストールされた）

4.3.4.2 管理対象ノードでの CMS 証明書のインストール

管理対象ノードに CMS 証明書をインストールするには、次の手順に従います。

1. *アップデートマネージャ*のメインウィンドウを開きます（図 18 を参照）。
2. *すべてのサーバ*で、CMS 証明書をインストールする管理対象ノードを選択します。
3. *アップデート詳細*タブの*アップグレード*ビュー（図 19 を参照）で、選択したノードの CMS 証明書に関するインストールオプションを示している行を選択します。
4. 管理対象ノードに CMS 証明書をインストールする新しいアップデートジョブを作成して開始します。

4.3.4.3 管理対象ノードからの CMS 証明書のアンインストール

管理対象ノードから CMS 証明書をアンインストールするには、次の手順に従います。

1. *アップデートマネージャ*のメインウィンドウを開きます（図 18 を参照）。
2. *すべてのサーバ*で、CMS 証明書をアンインストールする管理対象ノードを選択します。
3. *アップデート詳細*タブの*ダウングレード*ビューで、*新バージョン*列に「Unstinstall」と表示される行を選択します（66 ページ の図 22 を参照）。
4. 管理対象ノードから CMS 証明書をアンインストールする新しいアップデートジョブを作成して開始します。
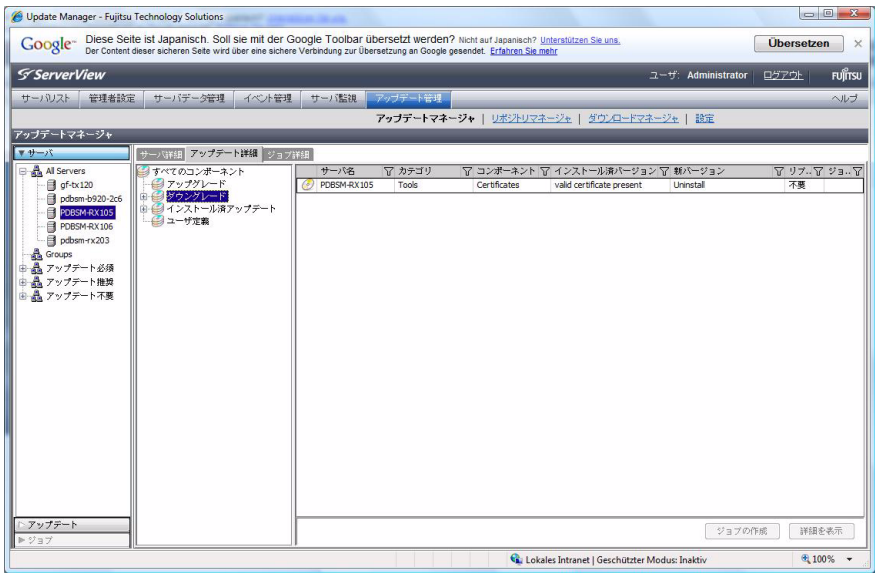
図 22: アップデートマネージャのメインウィンドウ - 「アップデート詳細」タブ（「ダウングレード」ビュー）

5 Operations Manager へのアクセスに関する役割ベースの許可

この章では、ユーザ役割 *Administrator*、*Operator* および *Monitor によって付与される権限に関する詳細情報を提供します。*

「禁止されている」コンポーネントは、Operations Manager GUI で無効になります。

ユーザに割り当てられているロールに必要な許可でない場合、Operations Manager GUI に表示されるコンポーネントおよび機能は非アクティブ（グレー表示）になります。図 23 に、例として Operations Manager の開始ウィンドウを示します。

図 23: Operations Manager 開始ウィンドウ - 「グレー表示」されるコンポーネントは許可されません

以降の項では、以下の個々について詳しく説明します。

– 69 ページ の「Operations Manager の開始ウィンドウ」
– 70 ページ の「Operations Manager GUI のメニューバー」
– 72 ページ の「サーバリスト」
– 73 ページ の「単一システムビュー」
– 74 ページ の「アラームモニタ」
– 75 ページ の「アップデートマネージャ」
– 75 ページ の「RAID Manager」

5.1 Operations Manager の開始ウィンドウ

開始ウィンドウには、Operations Manager のすべてのコンポーネントが表示されます。

認証の詳細	ユーザ役割で許可		
	Administrator	Operator	Monitor
サーバリスト			
ServerList へのアクセス（すべてのシステムの単一システムビュー（73 ページを参照）へのアクセスを意味します）	X	X	X
管理者設定			
ServerBrowser へのアクセスと検索の実行	X	X	
Server Configuration へのアクセス（Server Configuration Manager）	X		
*基本設定ウィザード*へのアクセス	X		
サーバデータ管理			
Archive Manager へのアクセス	X	X	
Inventory Manager へのアクセス	X	X	
イベント管理メニュー			
*アラームモニタ*へのアクセス（詳細は 74 ページを参照）	X	X	X
*アラーム設定*へのアクセス	X		
Threshold Manager へのアクセス	X	X	
監視メニュー			
Performance Manager へのアクセス	X	X	
Power Monitor へのアクセス	X	X	X
アップデート管理メニュー			
*アップデートマネージャ*へのアクセス	X	X	
*リポジトリマネージャ*へのアクセス	X		
*ダウンロードマネージャ*へのアクセス	X		
Configuration へのアクセス : *Update Configuration* へのアクセス許可	X		

表 3: 役割ベース認証 : Operations Manager の開始ウィンドウ

5.2 Operations Manager GUI のメニューバー

メニューバーには、Operations Manager のすべてのコンポーネントが表示されます。

認証の詳細	ユーザ役割で許可		
	Administrator	Operator	Monitor
ServerList メニュー			
ServerList へのアクセス（すべてのシステムの単一システムビュー（73 ページを参照）へのアクセスを意味します）	X	X	X
アーカイブのインポート（*ServerList - Import Archive*）	X	X	
ServerList のインポート（*ServerList - Import Server*）	X	X	
ServerList のエクスポート（*ServerList - Export Server*）	役割不要。		
ServerList 設定の定義			
管理者設定メニュー			
Server Browser にアクセスして検索を実行	X	X	
Server Configuration へのアクセス（Server Configuration Manager）	X		
User Passwords : ユーザパスワードの追加 / 変更 / 削除	X		
National settings、*基本設定ウィザード* : デフォルト値の変更	X		
Asset Management メニュー			
Archive Manager へのアクセス	X	X	
Inventory Manager へのアクセス	X	X	
イベント管理メニュー			
*アラームモニタ*へのアクセス（詳細は 74 ページを参照）	X	X	X
アラーム設定へのアクセス	X		
Threshold Manager へのアクセス	X	X	
MIB Integrator へのアクセス	X	X	
監視メニュー			
Performance Manager へのアクセス	X	X	
Power Monitor へのアクセス	X	X	X

表 4: 役割ベースの認証 : Operations Manager GUI のメニューバー

認証の詳細	ユーザ役割で許可		
	Administrator	Operator	Monitor
アップデート管理メニュー			
*アップデートマネージャ*へのアクセス	X	X	
*リポジトリマネージャ*へのアクセス	X		
*ダウンロードマネージャ*へのアクセス	X		
Configuration へのアクセス : *Update Configuration* へのアクセス許可	X		

表 4: 役割ベースの認証 : Operations Manager GUI のメニューバー

5.3 サーバリスト

認証の詳細	ユーザ役割で許可		
	Administrator	Operator	Monitor
ServerList へのアクセス（すべてのシステムの単一システムビューへのアクセスを意味します）	X	X	X
管理対象ノードの変更。*新規サーバの追加*、*Copy to group*、*削除*のための権限。	X	X	
選択したサーバ、サーバグループ、*すべてのサーバ*での電源操作の実行。	X	X	
Delete archives（選択したサーバのみ）と *Archive Now*（サーバグループまたは *すべてのサーバ*も対象）	X	X	
アラーム設定の変更。*Enable / Disable muted mode*（選択したサーバ、サーバグループ、または *すべてのサーバ*）	X		
Clear Alarms、*Clear All Alarms*（*すべてのシステム*またはグループのみ）	X	X	
接続テストの実行	X	X	X
リモート管理ツールの起動	X	X	
ノードの検索（サーバなど）およびサーバのブラウザへのアクセス	X	X	

表 5: 役割ベース認証 : サーバリスト

5.4 単一システムビュー

単一システムビューは通常、ServerList から呼び出します。

認証の詳細	ユーザ役割で許可		
	Administrator	**Operator**	**Monitor**
単一システムビュー - 関連する認証			
ServerList へのアクセス（すべてのシステムの単一システムビューへのアクセスを意味します）	X	X	X
*リモート管理*へのアクセス	X	X	
PrimeCollect へのアクセス	X	X	X
*オンライン診断*へのアクセス : 選択したサーバでオンライン診断を開始	X	X	
Delete archives（選択したサーバのみ）と *Archive Now*（サーバグループまたは *すべてのサーバ*も対象）	X	X	
システムの LED の検出のオン / オフ	X	X	
単一システムビュー - パフォーマンス管理			
Threshold Manager へのアクセス	X	X	
単一システムビュー - ブートオプション			
選択したサーバでの電源操作の実行（オン / オフ / 再起動）	X		
単一システムビュー - プロパティ			
接続テストの実行	X	X	X

表 6: 役割ベース認証 : 単一システムビュー

5.5 アラームモニタ

認証の詳細	ユーザ役割で許可		
	Administrator	Operator	Monitor
ServerList へのアクセス（すべてのシステムの単一システムビューへのアクセスを意味します）	X	X	X
サーバ関連のコンテキストメニュー : *Copy to group...*、*Rename*、*Server Properties*	X	X	
サーバ関連のコンテキストメニュー : 新規サーバの追加（ノードの検索（サーバなど）およびサーバのブラウザへのアクセス）	X	X	
サーバ関連のコンテキストメニュー : *Power Management* : 電源操作の実行（システムの電源のオン / オフ / 再起動）	X	X	
サーバ関連のコンテキストメニュー : *Archive now*（サーバグループまたは*すべてのサーバ*でも）	X	X	
サーバプロパティの変更	X	X	
アーカイブの削除（選択したサーバのみ）	X	X	

表 7: 役割ベース認証 : アラームモニタ

5.6 アップデートマネージャ

認証の詳細	**ユーザ役割で許可**		
	Administrator	**Operator**	**Monitor**
アップデートマネージャへのアクセス	X		
Update Configuration の追加 / 変更	X		
ジョブの削除	X		
リリースされたジョブの削除	X	X	
管理対象ノードでのアップデートエージェントデータのクリーンアップ	X		
ファームウェア / ソフトウェアアップデートを使用するジョブのコピー	X		
ファームウェア / ソフトウェアアップデートを使用するジョブの作成	X		
ジョブのリリース	X		

表 8: 役割ベース認証 : アップデートマネージャ

5.7 RAID Manager

認証の詳細	**ユーザ役割で許可**		
	Administrator	**Operator**	**Monitor**
RAID Manager ャへのアクセス	X	X	X
RAID Configuration の追加 / 変更	X	X	

表 9: 役割ベース認証 : RAID Manager

索引